Temperature Control System

Luke Longo and Kevin O'Hare
Rowan University

December 2, 2018

1 Design Overview

The overall objective of Milestone 2 is to use our newly gained knowledge and experience from this semester to implement a closed loop system. This system will be able to bring together three different systems being an MSP430 micro-controller, temperature sensor, and a fan all into one cohesive closed loop system. A breadboard circuit is constructed that has a voltage regulator, and the temperature sensor touching each other. The reason for this is the voltage regulator is solely used for generating heat, and the temperature sensor will read the heat coming off it. First, the MSP430 micro-controller will take this value and convert it from analog to digital. We will take that value and convert it back to analog in the code so it can be used to adjust the fan speed using pulse width modulation (PWM) to either cool down the regulator or allow it to heat up. The temperature values will be read and set through RealTerm. When a value is sent through RealTerm our system is designed to bring the temperature to that value and have it hold that temperature with a window of $\pm 3°$C.

1.1 Design Features

The design features include:

- UART Communication
- PTAT Temperature Sensing
- ADC Conversion
- Fan controlled using PWM
- Bang Bang Control Scheme

1.2 Featured Applications

Possible applications of temperature control systems include:

- Computer Fan Control
- Room Temperature Control
- Prevention of Rapid Heating for Ceramic Firing Process
- Energy-saving Measure for Solder Tank Idling
- Redundancy in Temperature Control and Temperature Monitoring
- Error Detection in Control Loops

1.3 Design Resources

The code for this assignment is located on github.com. The link to our repository is:
`https://github.com/RU09342-F18/introtoembedded-f18-milestone2-art-thow-feeling-it-now`
`blob/master/Code/main.c`

1.4 Block Diagram

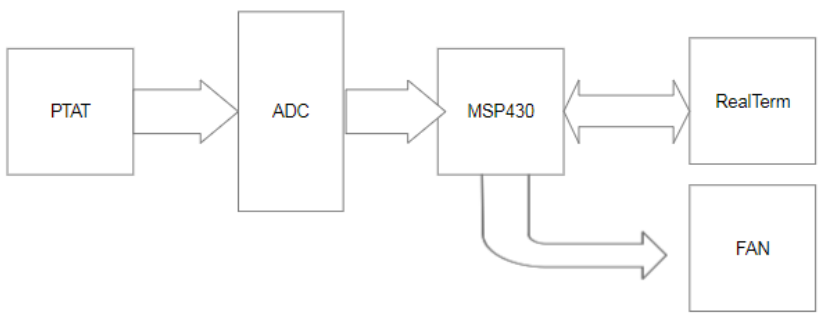

Figure 1: Simple Block Diagram

1.5 Board Image

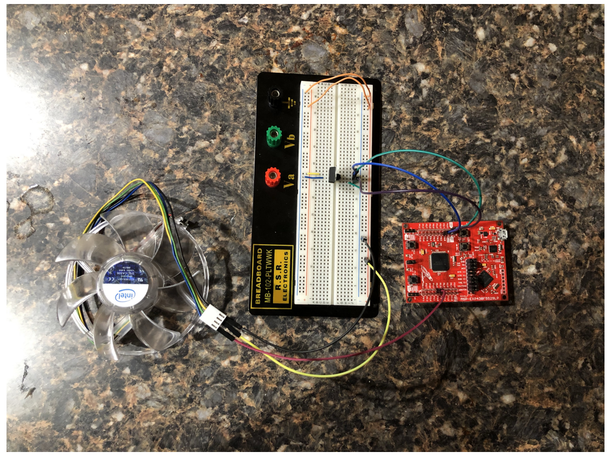

Figure 2: Board Setup

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Micro Controller	MSP430F5529	Micro controller required for code
PTAT	LM60 CIZ 2.7 V	Temperature Sensor
Voltage Regulator	5 V 1 A	Voltage regulator used for ADC
Intel Computer Fan	12 V	Cooling of PTAT and voltage regulator

3 System Description

This closed loop system is designed to read temperature and be able to adjust the fan speed accordingly to bring the temperature to a desired value.

Initially when a voltage is applied to the board the voltage regulator will begin to heat up, and the temperature sensor (PTAT) touching this regulator will read its temperature. The PTAT outputs a voltage as a function of the temperature it experiences. This voltage passes through an ADC conversion and is converted into a digital value that can be stored in a register, for our purposes ADC12MEM0. Then within the code the digital value is converted back to an analog value so that it may be used to find the temperature being read by the PTAT. The following equations are used to find temperature.

$$VoltageIn = \frac{(ADC12MEM0 * 3.3)}{4095} \tag{1}$$

$$Temperature = (VoltageIn - 0.424) * 160 \tag{2}$$

The temperature value that is calculated is stored as the current temperature, and is displayed on our terminal.

In our design we use two timer capture/compare registers (TAxCCRy) TA0CCR0, and TA0CCR. TA0CCR0 is used to set the limit of the PWM to 1000 and TA0CCR1 is used to control the fan speed. Using the OUTMOD function of Timer A, when the timer reaches the values of TA0CCR1, the GPIO pin controlling the fan speed will reset, and when the timer reaces the value of TA0CCR0 the GPIO pin will be set. This sequence of on then off allows the microcontroller to control the duty cycle of the pin, and therfore the PWM. The bang bang control method is used to controll the fan speed, and it works as follows:

- Current Temp $>$ Set Temp + 2: The TA0CCR1 will be incremented to increase PWM, and therefore increase fan speed.
- Current Temp $<$ Set Temp - 2: The TA0CCR1 will be decremented to indecrease PWM, and therefore decrease fan speed.

This method will allow the fan to be able to correct itself when the temperature goes over the set temperature or below the set temperature. This method has our temperature of our sensor is typically within the $\pm 3^{\circ}$ C threshold.

3.1 Detailed Block Diagram

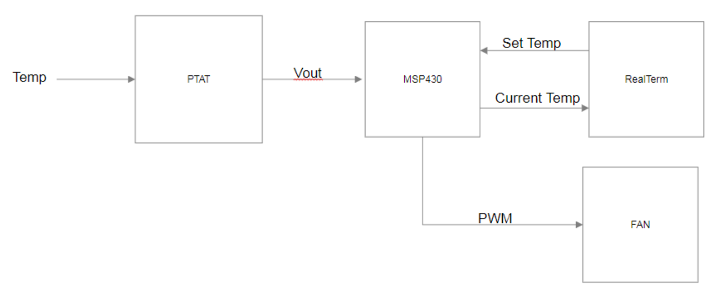

Figure 3: Detailed Block Diagram

3.2 Highlighted Devices

1. **MSP430F5529** - The MSP430F5529 (referred to as the F5529) is a micro controller manufactured by Texas Instruments. It is used for its timer module, GPIO pins, ADC conversion, and its UART mode. The timer module Timer A0 is capable of setting or resetting GPIO pins based on values in its Capture/Compare registers (referred to as CCR). When the timer count reaches a value stored in a CCR, an action will be performed. The ADC conversion will take analog values

and convert them into digital values so they can be used. UART mode allows the micro controller to receive and transmit data via RX and TX pins respectively. We also use the RX interrupt service routine to perform action upon receiving a byte on the RX line.

2. **PTAT** - The PTAT is a temperature sensor that is used to read temperature off the voltage regulator. The PTAT will output a voltage value that is used to find the temperature through calculation within the code. In our design the PTAT was used to read temperature coming off of our 5V regulator.

   The equation for the PTAT's voltage output is as follows:

   $$Vout = 6.25\frac{mV}{^{\circ}C} \cdot Temp(^{\circ}C) + 424mV \tag{3}$$

3. **Voltage Regulator** - The 5V regulator is used as a heat source in our design. When a large voltage source is applied to it, it will take that voltage and decrease or regulate it. In our case we applied 12V to it so it would power the fan and heat up, the output of the regulator brought the voltage down to about 6V.

4. **Intel Fan** - This fan is a basic intel fan that is used for cooling in computers. The fan has three useful leads that were used: Black (Ground), Blue (PWM), Yellow (Power). The power initially required to power the fan is 12 Volts.

4 SYSTEM DESIGN THEORY

4.1 Design Requirements

Our design revolves around heating up a voltage regulator, and having a fan control its temperature. It is important that our design has an input of at least 12 V because the fan needs 12 V to be turned on, and the voltage regulator requires at least 5V to begin to heat up. A load for the regulator output is required in order to draw current, as the regulator will not heat up if there is no current. The particular load used must be picked carefully. It must not draw a current larger than 1 A, as the is the max output of the regulator. Smaller loads are dangerous, and could cause the power supply to limit itself to prevent damage.

5 Getting Started/How to use the device

In order to get started on this project you will need to download the following software:

1. Code Composer Studio
2. Realterm

Code Composer Studio is a program that will allow you to directly write code to your board and test it all in one window. Realterm is a program that is used as a terminal that allows you to send bytes of information to your microprocessor.

To implement the project you will need the following equipment:

1. Breadboard
2. Jumpers
3. Power Resistor (or similar appropriate load)
4. DC Power Supply
5. DMM Probes
6. PTAT (Temperature Sensor)
7. Voltage Regulator (1 A Output)

The voltage regulator is placed on the board to generate heat, and the temperature sensor reads the heat of the regulator. The power resistor is used as a load for the regulator.

6 Getting Started Software/Firmware

The software you will need for this device is Code Composer Studio (CCS) and Realterm. CCS is used to create the program run by the micro processor. After a project is opened in CCS, it can be built for debugging by clicking the hammer icon on the toolbar. Then the program can be debugged by pressing the bug icon. This programs the micro controller with the program.

6.1 Communicating with the Device

Realterm is the preferable method of communicating with the F5529 as it is able to send bytes of data at a time. To setup up Realterm two things must be chosen: the desired baud rate, and the COM port linked with the processor. The baud rate for this device is 9600, but may be changed if necessary by tweaking the UART peripheral. The COM port can be found by connecting the microssor the you computer and checking your Device Manager (System Information for Mac users). Under Ports will be listed a COM port likely labled "UART". This is the COM Port you will be using for Realterm.

Now that we know the baud rate and the COM Port, open Realterm and select the Port tab. Set the baud rate and COM Port, and open the connection. If you cannot find the COM Port in the drop-down list, you can double click the box to scan for ports. With these set, select the Send tab. From here you can enter a value to set the temperature.

7 Test Setup

Testing this system can be done using the CCS Debug mode to run the code. When running the code the Realterm terminal should be open to be able to look at the displayed temperature values. Also the output of the PTAT temperature sensor should be connected to an oscilloscope to monitor the oscillating temperature values. Specifically this should be used to make sure that the temperature remains within the $\pm 3^{\circ}$ C.